UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/880,888	06/15/2001	Yaron Kashai	V02/9	6626

7590 07/14/2004

DR. D. GRAESER LTD.
C/O THE POLKINGHORNS
9003 FLORIN WAY
UPPER MARLBORO, MD 20772

EXAMINER
VO, TED T

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 07/14/2004

9

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/880,888	**Applicant(s)** KASHAI ET AL.	
	Examiner Ted T. Vo	**Art Unit** 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 April 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed on 04/23/2004, where Applicants have filled:

- Amending the Drawings in replying to the Drawing Objection by Examiner; the new Drawings are accepted by Examiner for exanimation purpose only.

- Substituting a new Oath/Declaration for the pervious defective Oath/Declaration.

- Amending the specification in replying to the objection by Examiner; the amendment results withdrawing the specification's objection.

- Claims 6 and 15 are amended. With respect to the amendment of Claims 6 and 15, the rejection under 35 USC 112 second paragraph is withdrawn.

2. Claims 1-22 are pending in the application.

Claims 1-22 stand finally rejected under 35 USC 102(e) as being anticipated by Edwards et al., (US No. 6,625,797).

Response to Arguments

3. Applicant's arguments in the Remarks filed on 4/23/04 have been fully considered. Applicants brief broadly the teaching of Edwards (re: Remarks Page 11), and generally argue differences between their Claims and Edwards without referring their claimed limitation or addressing the reference's teaching cited in the previous Office Action. These arguments would not be persuasive.

For example, in Remarks:

Started at "By contrast..." in page 11 through page 12, Applicants discuss generally without addressing directly to their claimed limitations and Office action's citations.

In page 13, at lines 5-6, Applicants stated, "*Edwards does not teach a method for compiling languages which feature "structure constraints"*".

Examiner responds: "*structure constraints*" as argued by Applicants is recited in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

In page 13, at lines 7-14, Applicants stated, "*Edward gives examples such as C++ and Java, which are high-level software languages. High-level software languages do not feature constraints. These software are not equivalent to verification language*".

Examiner responds: "*High-level software languages"* in the reference is not cited in office action citation. Edwards: column 5, lines 51-53, high-level <u>source specification</u> is referring to *the first language features a hierarchy of objects,* recited the preamble. Moreover, there is no such limitation *verification language* in the claims as argued. In fact, Edwards teaches this preamble's recitation <u>*first language features a hierarchy of objects*</u>, with regards to high-level <u>source specification</u> (Edwards: column 5, lines 51-53). This source specification is a compiled version, which is included with analysis, annotation, control and data flow graph, user preferences and <u>constraints</u>, as shown in Figure 1. This type of source specification is no longer C++ or Java high-level languages as contended by Applicants. This specification is translated again into hardware representation (Claimed limitation: second language).

In page 13, at lines 7-14, Applicants stated "*Edwards fails to show how to determine which objects instance exist in the system for a program being compiled*".

Examiner responds: In the claims, particularly, in the independent Claim 1, there is no such limitation, or featured as "*how to determine which objects instance exist in the system for a program being compiled*". In fact, the reference also teaches this argument, "*how to determine object instance exist in the system*" (Edwards: see Column 6, lines 11-14: "each bytecode is now defined to causes the instantiation of a specific hardware circuit in the final hardware implementation. This is accomplished through a two step compilation-translation process"). It is noted that bytecode is an element as a node in Control and Data flow Graph phase (Figure 1).

In page 14, at lines 3-6, Applicants stated "*the method of the present invention is able to handle the complexity arising from multiplicity of objects, for example by determining the elaborated roles (storage), deriving the access pattern (creation of the conflict graphs), conflict resolution and scheduling*".

Examiner responds: There is no such claimed limitation in the claims.

In page 14, at lines 12-16, Applicants merely contend "*The static framework of resource in claim 1 is not equivalent to hardware-implementation independent flowgraph*" and also merely contend "*substituting nodes in the graph of Edwards is not equivalent or even similar to mapping dynamic behavior of the second language in Claim 1*".

Examiner responds: The flowgraph of nodes is known for "static", wherein these nodes represent the bytecodes, and are included with annotation (dynamic behavior) (See in column 6, lines 15-42). Edwards references the flowgraph as Generation of the hardware-implementation independent flowgraph (lines 15-16). The broad limitation "static framework of resource" could be read by Generation of the hardware-implementation independent flowgraph. The annotations in a node detail the specific hardware to replace for the node (See Column 3, 21-35), and bytecode represents the node (emphasis added) (Claimed limitation: *supporting the dynamic behavior of the code written in the first language*). Edward teaches a second phase that substitutes a hardware circuit for each node (See Column 6, lines 26-27) that reads the functionality of *mapping dynamic behavior of the second language* (hardware implementation/hardware circuit). Since Applicants are not able to explain the differences, but merely asserting, "not equivalent", such arguments are not persuasive.

The rejection of Claims 1-22 under 35 U.S.C. 102(e) as being anticipated by Edwards will be maintained.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-22 are rejected under 35 U.S.C. 102(e) as being anticipated by Edwards et al., (US No. 6,625,797).

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per Claim 1:

Edwards discloses a method for translating a high-level source specification [claimed language: 'first language'] such as a compiled version of a source code (see Figure 1) into a target language [claimed language: 'second language'] called hardware representation (see column 5, lines 39-44). The method includes generation of nodes that represent hardware circuits (resources) by analyzing the source specification (see column 5, lines 55-61). Nodes and control paths [claimed language: 'static frame work'] represent bycode, functions, annotations (dynamic behavior) from the source specification (see column 3, lines 36-50).

The translation includes generation of a hardware-implementation independent flowgraph [claimed language: 'static framework'] with nodes, where each node represents a distinct hardware circuit (see column 6, lines 15-16). Then the nodes are substituted [claimed language: 'mapping'] with hardware circuits (see column 6, lines 26-32) based on the semantics of bytecodes which create the nodes [Nodes with a specified semantic: 'underlying control structure'] for representing the target language (see column 5, lines 39-44).

Despite the terminological differences, the teaching covers the claim limitations:

"*A method for at least semi-automatically translating code written in a first language to a second language featuring constraints and featuring dynamic behavior, wherein the first language features a hierarchy of objects, such that relationships between the objects are determined by constraints, the method comprising:*

*detecting an underlying control structure for code in the first language***"** (see column 5, lines 51-53, 'The information gathered [*detecting*] from the bytecodes/source code [*code in the first language*] is obtained through analysis [*detecting*] of the language semantics and structure [*underlying control structure*]. See column 6, lines 27-29, 'The specific function implemented in each hardware circuit is specified by the semantics of the bytecode which created that node': Edwards teaches obtaining bytecodes which are specified by bytecodes' semantics and creating a node from an obtained bytecode. This has means of *detecting an underlying control structure*).

"*creating a static framework* (hardware-implementation independent flowgraph, nodes) *of resources for supporting the dynamic behavior* (See Column 3, 21-35, 'annotations', see column 6, lines 7-14, bytecode) *of the code written in the first language***"** (see column 6, lines 15-25, referring to: "generation of the hardware-implementation independent flowgraph [*static framework*]. Each bytecode in compiled source code language [*in the first language*] specifies a node [*supporting the dynamic behavior*] to be inserted into the initial flowgraph", and "This node [*static framework*] represents a distinct hardware circuit [*resources*]'); *and*

"*mapping the dynamic behavior* (See column 3, lines 21-35: "Nodes may be annotated with supplementary information.... For example, these annotations may include, but are not limited to:latency(...), gate depth(...), speed..", see column 6, lines 37-42, flow graph annotations, bytecode: *dynamic behavior*) *to the second target language* (hardware representations/circuits) *according to said static framework* (nodes, data, and control flow) *of resources and said underlying control structure*" (see column 6, lines 26-42, supplied constraints and preferences and semantics of the bytecode which created that node : *underlying control structure*').

As per Claim 2: Edwards discloses, "*The method of claim 1, wherein said underlying control structure is detected by control flow analysis for an action* (see action in claim 1, step 'detecting...'), *to determine at least one of a condition and a trigger for causing said action to execute*"

(see column 7, lines 25-39, referring to: 'In this case of a conditional fork, one branch [*determine at least one of a condition*] will represent the bytecodes that would be executed on a "true" result [*trigger for causing said action to execute*]').

As per Claim 3: Edwards discloses, "*The method of claim 2, wherein said at least one of the condition and trigger is a guard for governing execution of said action*" (see action in claim 2),

"*such that control flow analysis comprises at least determining at least one guard for each action*" (see column 7, lines 25-39, referring to: "True" result [*determining at least one guard*]).

As per Claim 4: Edwards discloses, "*The method of claim 3, wherein said control flow analysis further comprises determining a plurality of guards for creating a sequential control flow graph*" (in light of the specification [in the specification: page 15, line 7: 'guard nodes represent a branch point'], see column 7, lines 25-39, referring to: 'one branch [*a guard*] will represent the bytecodes that would be executed on a "true" result, the other branch [*a guard*] represents the bytecodes which would be executed on a "false" returned from the conditional operation [*sequential control flow graph*], and 'data between the sequenced operation').

As per Claim 5: Edwards discloses, "*The method of claim 4, wherein said underlying control flow for the code in the first language is detected by:*

parsing the code; and creating an abstract syntax tree" (see figure 2; referring to: 'Parse', 'Annotated CDFG'. For a particular abstract syntax tree: see column 8, lines 41-54, 'a series of algorithms may be applied to each flowgraph and to the collection of flowgraphs...');

"*wherein said at least one node of said sequential control graph retains a reference to a node of said syntax tree*" (see column 5, lines 39-44, 'library reference, etc., according to the annotation of each node and path [*retains a reference to a node of said syntax tree*]').

As per Claim 6:

Edwards discloses, "*The method of claim 4, wherein said control flow analysis comprises: computing a trigger structure of each process of the code in the first language*" (see column 7, lines 25-39, 'True result' and 'false result', and 'sequenced operation').

"*determining said at least one guard for each action of said process*" (see column 7, lines 25-39, '"True" result' and '"false" result', have means for determining guards'); *and*

"*determining a segmentation of said process into a plurality of segments*" (see column 16, lines 4-7, 'wait() method indicates that the control flow through a specific segment [*determining a segmentation*] of the flowgraph', and see column 9, lines 42-52, 'inserting registers [*plurality of segments*] (sequential elements) into the flowgraph': Examiner note: in light of the specification, figure 9, an insertion of registers has means of a plurality of segments).

As per Claim 7: Edwards discloses, "*The method of claim 6, wherein said control flow analysis further comprises: unrolling at least one loop*" (see column 5, lines 6-9, 'loop unrolling')

As per Claim 8: Edwards discloses, "*The method of claim 6, further comprising retiming at least one action*" (and see column 5, lines 19-26, 'Physical memory and routing elements such as register, flip-flop, and multiplexers are inserted into the data flow in order to ensure their arrival times in each destination instructions [*retiming*]'; see column 9, lines 42-52, 'inserting registers (sequential elements) into the flowgraph'. This insertion is further discussed by Edwards for fixing timing, in which the flowgraph generation does not fulfill, by breaking the control flow and inserting the registers for delaying the same clock cycles: See column 9, lines 47-52, 'Each time a register is inserted between two nodes [*retiming*] in the flowgraph, the control signal between those same two nodes is broken and a flip-flop inserted. This has the effect of delaying [*retiming*] the control by the same number of the cycles as the data').

As per Claim 9:

In light of the specification (page 15, lines 5-12),

Edwards discloses, "*The method of claim 6, wherein each node is selected from a group consisting of a basis node for representing a list of non-timing consuming action* (see column 9, lines 45-52, referring to "nodes" in 'two nodes the graph', and 'same two nodes. See column 4, lines 11-12, 'Node'), *a guard node for representing a branch point* (see column 7, lines 25-39, referring to: 'one branch' and 'other branch');

and a wait node for containing a temporal expression" (In column 16, lines 4-25 it discusses the wait() implements a node (line 16) [wait node] at the broken point where the wait() method is called (lines 6-7)).

As per Claim 10: Regarding the limitation, "*The method of claim 2, wherein said control flow analysis detects at lest one malformed control structure for manual alteration by a user*"

Edwards discloses this limitation by providing an insertion during compilation and translation (see figure 1), where a user can insert user preferences and constraints by using "pause" (see column 16, lines 39-67).

As per Claim 11: Edwards discloses, "*The method of claim 1, wherein said static framework of resources is created by elaboration, wherein elaboration is performed by allocating a sufficient number of state holding elements for representing dynamic behavior of the code written in the first language*" (see column 5, lines 39-67, 'The fully resolved, elaborated...'; and column 6, lines 4-42, discusses the full elaboration).

As per Claim 12: Edwards discloses, "*The method of claim 11, wherein said state of holding elements are allocated by:*

recursively analyzing a structure of the code written in the first language (see column 5, lines 30-35, 'recursively resolved...');

determining structural constraints; (see column 16, lines 45-50, 'these structures may (selectively via user constraints) include:');

creating an elaborating graph from said structure of the code in said structural constraints" (see Figure 1, the flow between 'User Preferences and Constraints' and Control and Data Flow Graph').

As per Claim 13: Edwards discloses, "*The method of claim 12, wherein said elaboration graph features a plurality of nodes selected from group consisting of scalar nodes, struct nodes and list nodes*" (see column 4, line 11 the definition of Node; see column 6, lines 15-25, 'the node represents a distinct hardware circuit in the resulting the hardware implementation', and 'Node's characteristic... [*scalar nodes, struct nodes and list nodes*]', where *scalar nodes, struct nodes and list nodes* are inherent in the semantics of the source program)

As per Claim 14: Edwards discloses, "*The method of claim 12, wherein said elaboration graph is unfolded to completely represent a plurality of structures of the first language*" (see column 5, lines 39-44, 'elaborated and annotated logic design flow graph representation [*represent a plurality of structures of the first language*]).

As per Claim 15: Edwards discloses, "*The method of claim 14, wherein the first language features symmetry* (see column 3, lines 51-56), *and wherein said elaboration graph is unfolded to overcome an asymmetrical feature of the code*" (see column 5, lines 19-26 'inserted into the data flows....so that the functionality of the original source is preserved [*overcome an asymmetrical feature*]'; see column 9, lines 40-52, 'inserting registers into flow graph while still maintaining correct functionality [*overcome an asymmetrical feature*]').

As per Claim 16: Edwards discloses, "*The method of claim 15, wherein said asymmetrical feature is selected from the group consisting of a temporal expression"* (see column 3, lines 51-58 'has the minimum sense of temporal operation [*consisting of a temporal expression*]' and see column 9, lines 40-52, 'inserting registers into flow graph [*asymmetrical feature*] while still maintaining correct functionality') "*and a time consuming method*" (see Figure 1: 'User preferences and constraints'. Examiner note: Involvement of a user in a process is *time consuming*).

As per Claim 17: Edwards discloses, "*The method of claim 12, wherein said underlying control structure is detected by control flow analysis for an action, to determine at least one of a condition and a trigger for causing said action to execute*"(see column 7, lines 25-39, 'The nodes, which correspond to operations' [*underlying control structure*], 'conditional operation' [*determine at least one of a condition*] and 'executed on a "true" result' [*trigger for causing said action to execute*]),

"where said at least one of a condition and trigger is a guard for governing execution of said action" (see column 7, lines 25-39, '"true" result'), *and*

"wherein a plurality of guards is determined for creating a sequential control graph" (see column 7, lines 25-39, '"true" result' and "false return from the conditional operation', and 'sequenced operation'),

"*the method further comprising:*

determining an interrelationship between said elaboration graph and said sequential control graph"(see column, 9, 12-39, 'for loops', "while loops' and do-while loops' [*determining an interrelationship*] [*said sequential control*]).

As per Claim 18: Edwards discloses, "*The method of claim 17, wherein said interrelationship is used to detect a race*" (see column 9, lines 29-30, 'This eliminates asynchronous feedback loops with undefined completion states [*detect a race*]').

As per Claim 19: Edwards discloses, "*The method of claim 17, wherein said interrelationship is used to compute an execution schedule*" (see column 9, line 11, 'Scheduling and Resource Sharing').

As per Claim 20: Edwards discloses, "*The method of claim 1, wherein the first language features at least one type of symmetry, and wherein said at least one type of symmetry is maintained when creating said static frame work of resources*" (see column 5, lines 19-26, 'Physical memory and routing elements such as register, flip-flop, and multiplexers [*static frame work of resources*] are inserted into the data flow in order to ensure their arrival times in each destination instruction, so that the functionality of the original source [*symmetry*] is preserved [*symmetry is maintained*]'; see column 9, lines 40-52, 'inserting registers into flow graph while still maintaining correct functionality [*symmetry is maintained*]').

As per Caim 21: Edwards discloses, "*The method of claim 1, wherein the first language is a verification language*" (see column 1, lines 9-13, 'high-level software language [*verification language*]' and see column 2, lines 25-30, 'high-level source specification [*verification language*]').

As per Claim 22: Edwards discloses, "*The method of claim 1, wherein the code, after translation to the second language, performs verification of a design* (see column 5, lines 39-44, annotated logic design flowgraph is translated into target language [*the second language*] ', and see column 1, lines 9-13, 'digital hardware implementations [*performs verification of a design*]').

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (703) 308-9049. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552.

The fax phone numbers:

(703) 872-9306 (for formal communication intended for entry);

(703) 746-5429 (for informal or draft communication, please label "PROPOSED" or "DRAFT").

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

TUAN DAM
SUPERVISORY PATENT EXAMINER

TTV
Patent Examiner
Art Unit: 2122
July 8, 2004